June 10, 2008

COMPUTER ENGINEERING DEPARTMENT

ICS 233

COMPUTER ARCHITECTURE & ASSEMBLY LANGUAGE

Final Exam

Second Semester (072)

Time: 7:30-10:30 AM

Student Name : _____

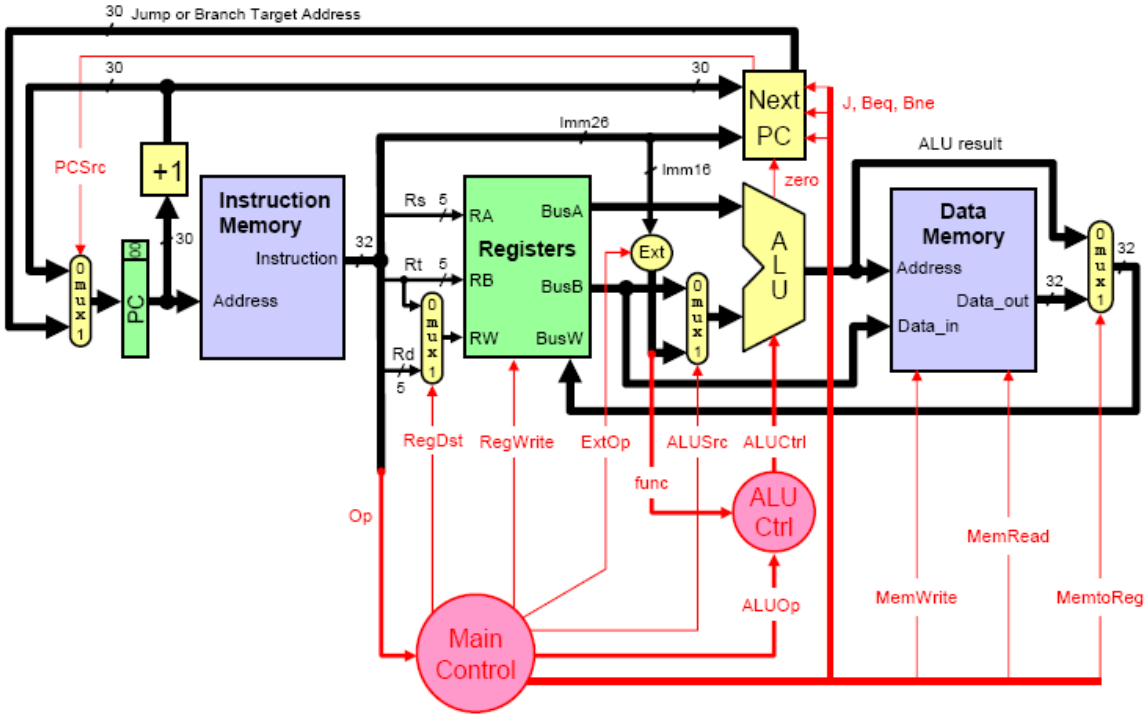
Student ID. : _____

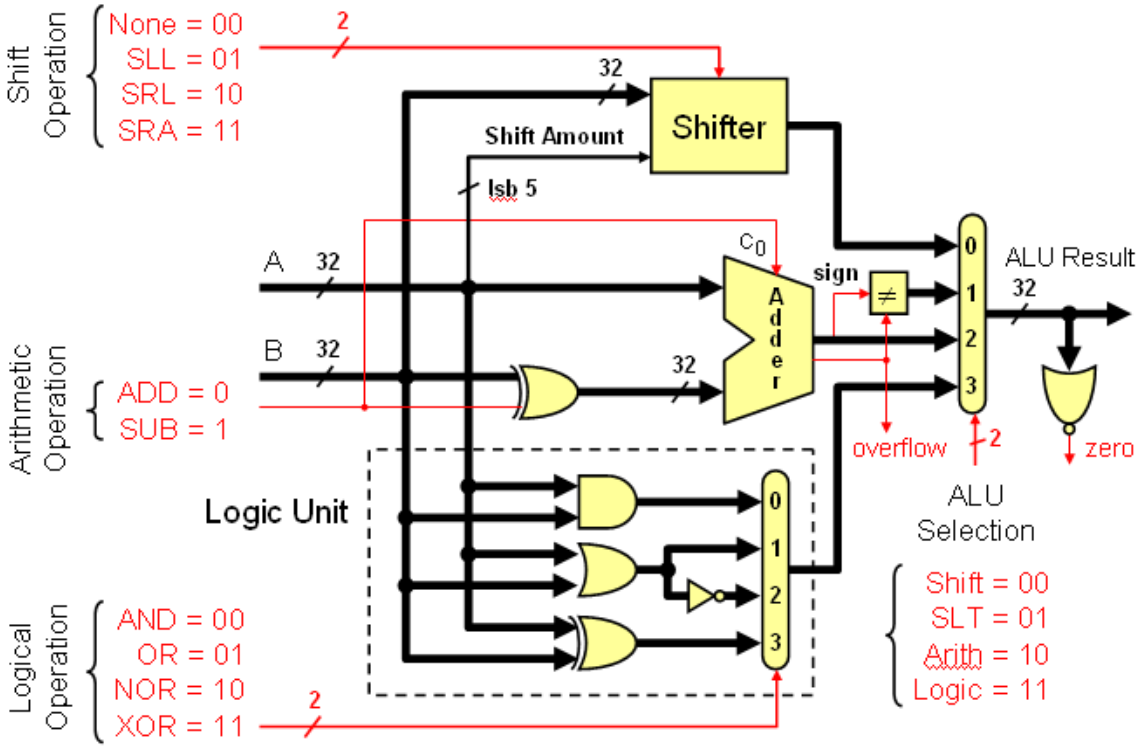
Question	Max Points	Score
Q1	30	
Q2	15	
Q3	15	
Q4	10	
Q5	18	
Q6	12	
Total	100	

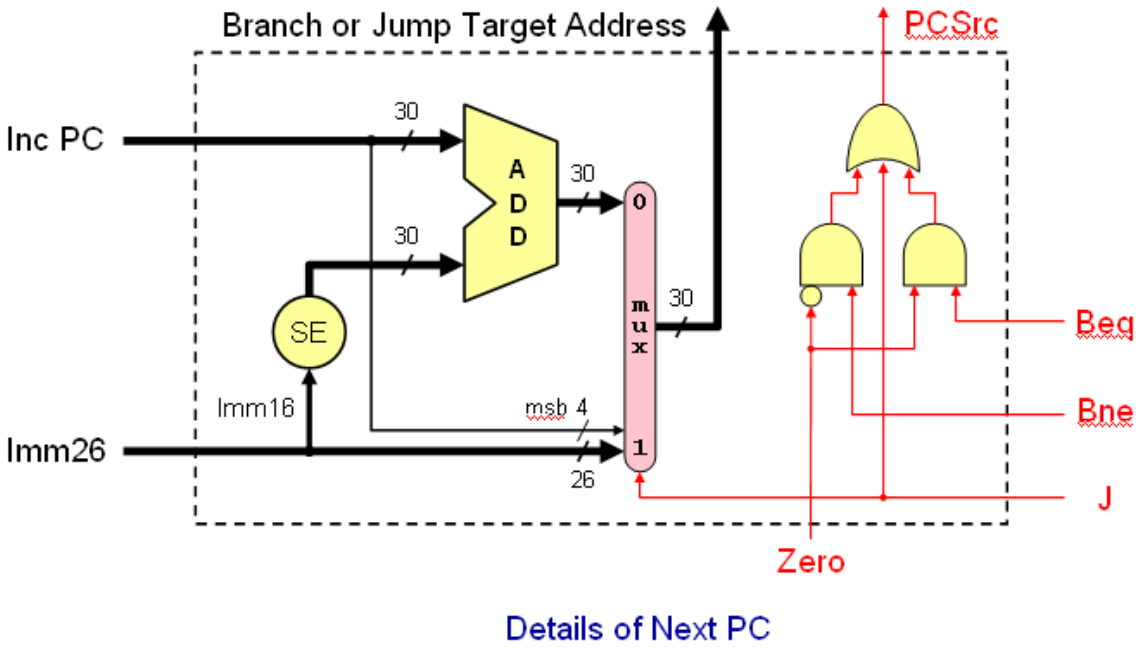
Dr. Aiman El-Maleh

[30 Points]

(Q1) Consider the single-cycle datapath and control given below along with ALU and Next PC blocks design for the MIPS processor implementing a subset of the instruction set:







Details of Next PC

(i) Show the control signals generated for the execution of the following instructions by filling the table given below:

Op	RegDst	RegWrite	ExtOp	ALUSrc	ALUOp	Beq	Bne	J	MemRead	MemWrite	MemtoReg
R-type											
addi											
xori											
lw											
bne											

The format of these instructions is given below for your reference:

Instruction		Meaning	Format					
sub	rd, rs, rt	rd = rs + rt	$Op^6 = 0$	rs^5	rt^5	rd^5	0	0x22
addi	rt, rs, imm^{16}	rt = rs + imm^{16}	0x08	rs^5	rt^5	imm^{16}		
xori	rt, rs, imm^{16}	rt = rs ^ imm^{16}	0x0e	rs^5	rt^5	imm^{16}		
lw	rt, imm^{16}(rs)	rt = MEM[rs+imm16]	0x23	rs^5	rt^5	imm^{16}		
bne	rs, rt, label	branch if (rs != rt)	0x05	rs^5	rt^5	imm^{16}		

(ii) We wish to add the following instructions to the MIPS single-cycle datapath. Add any necessary datapath modifications and control signals needed for the implementation of these instructions. Show only the **modified** and **added** components to the datapath. Show the values of the control signals to control the execution of each instruction.

a. lui

Instruction	Meaning	Format			
lui rt, imm^{16}	rt= imm^{16} << 16	Op^6 = 0xf	0	rt^5	imm^{16}

b. sltiu

Instruction	Meaning	Format			
sltiu rt, rs, imm^{16}	rt=(rs<imm?1:0)	Op^6 = 0xb	rs^5	rt^5	imm^{16}

c. bgtz

Instruction	Meaning	Format			
bgtz rs, label	branch if (rs>0)	Op^6 = 7	rs^5	0	imm^{16}

d. jalr

Instruction	Meaning	Format					
jalr rd, rs	rd=pc+4, pc=rs	op^6 = 0	rs^5	0	rd^5	0	9

(iii) Assume that the propagation delays for the major components used in the datapath are as follows:

- Instruction and data memories: 150 ps
- ALU and adders: 100 ps
- Register file access (read or write): 60 ps
- Main control: 20 ps
- ALU control: 20 ps

Ignore the delays in the multiplexers, PC access, extension logic, and wires.

a. What is the cycle time for the single-cycle datapath given above?

b. A friend of yours suggested modifying the MIPS instruction set architecture to remove the ability to specify an offset for memory access instructions. Specifically, all load-store instructions with nonzero offsets would become pseudo instructions and would be implemented using two instructions. For example: the instruction lw $t0, 4($t1) is implemented as:

```
addi  $at, $t1, 4          # add the offset to a temporary
lw    $t0, $at             # new way of doing lw $t0, 4($t1)
```

What will be the cycle time if the proposed simplified architecture were to be used? Under what conditions, the proposed architecture will be faster?

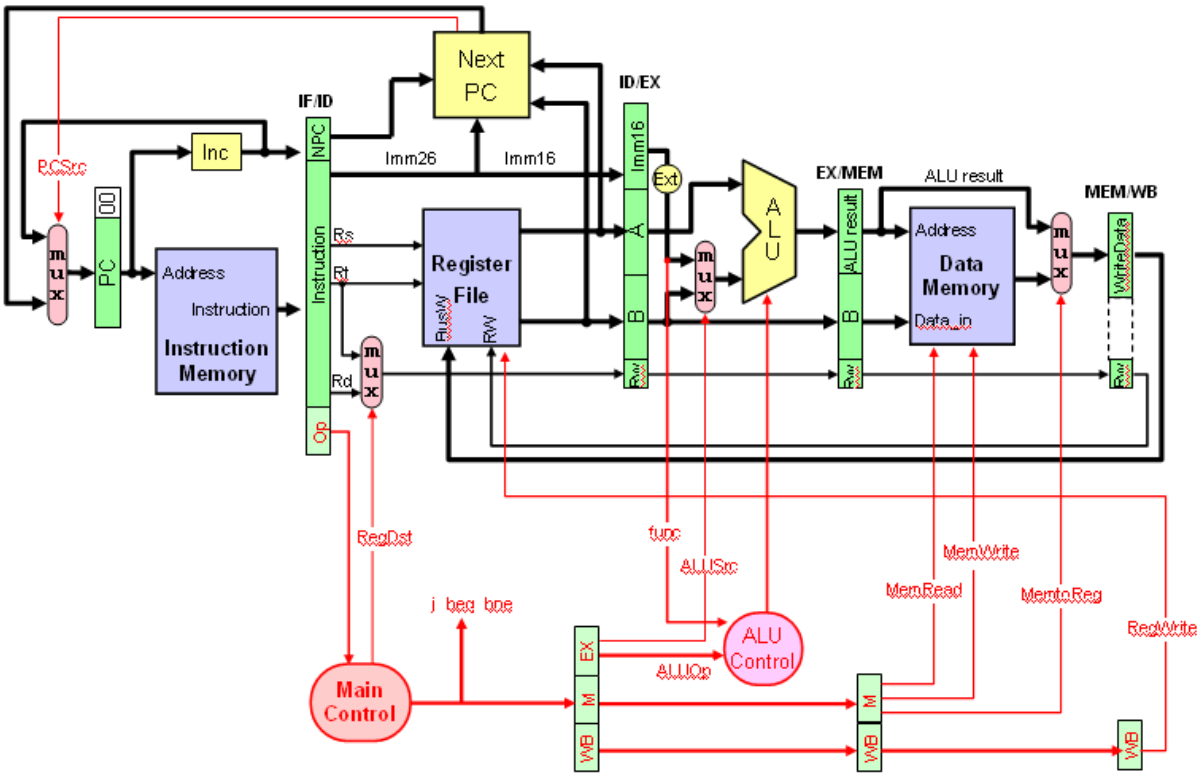
[15 Points]

(Q2) Consider the pipelined MIPS processor design given below:

(i) Make all the necessary changes to the given pipelined design to overcome data and control hazards by showing the design of hazard detection, forwarding and stall unit.

(ii) Show the control signals that will be used for forwarding along with their conditions.

(iii) Show the control signals that will be used for stalling the pipeline along with their conditions.



[15 Points]

(Q3) Consider the code given below:

```
add $1, $2, $3
lw $2, 8($1)
sub $2, $2, $1
sw $2, 8($1)
```

(i) Identify all the **RAW** data dependencies in the above code. Which dependencies are data hazards that will be resolved by forwarding? Which dependencies are data hazards that will cause a stall?

(ii) Using a multiple-clock-cycle graphical representation, show the instruction execution across the pipeline including forwarding paths and stalled cycles if any. How many clock cycles will be needed to execute the instructions?

[10 Points]

(Q4) Consider the MIPS program given below:

```
.data
Table: .word 2, 4, 3, 5, 6, 1
.text
.globl main
main:
        la      $a0, Table      # address of first element
        add     $a1, $a0, 20    # address of last element
max:    move    $v0, $a0        # max pointer = first pointer
        lw      $v1, ($v0)      # $v1 = first value
        move    $t0, $a0        # $t0 = array pointer
loop:   addi    $t0, $t0, 4     # point to next array element
        lw      $t1, 0($t0)     # $t1 = value of A[i]
        slt     $at, $v1, $t1   # if (A[i] ≤ max) then skip
        beq     $at, $0, skip
        move    $v0, $t0        # found new maximum
        move    $v1, $t1
skip:   bne     $t0, $a1, loop  # loop back if more elements
ret:
```

(i) Show the outcomes of each of the conditional branch instructions due to the execution of the program (T for taken, N for not taken).

(ii) List the predictions and the accuracies for each of the following dynamic branch predictions schemes:

a. 1-bit prediction, initialized to **predict not taken**.

b. 2-bit predictor, initialized to **weakly predict not taken**.

[18 Points]

(Q5) Assume that you have a cache with **64 bytes data size** (i.e. not including tag and valid bits). Consider the following series of address references given as 16-bit addresses:

0x00c2, 0x00c3, 0x00c4, 0x00c5, 0x0ec2, 0x0ec3, 0x0ec4, 0x0ec5, 0x00c2, 0x00c3, 0x00c4, 0x00c5, 0x00c6, 0x00c7, 0xffc6, 0xffc7.

(i) Assuming that the cache is organized as **direct-mapped** with **2-byte block size**, determine the number of bits in the offset, index and tag fields. Starting with an empty cache, show the offset, index and tag (**in binary**) for each address reference in the list and indicate whether it is a hit or a miss. What is the miss ratio for this sequence on this cache?

Offset = Index = Tag=

Address	Tag	Index	Offset	Hit/Miss
0x00c2				
0x00c3				
0x00c4				
0x00c5				
0x0ec2				
0x0ec3				
0x0ec4				
0x0ec5				
0x00c2				
0x00c3				
0x00c4				
0x00c5				
0x00c6				
0x00c7				
0xffc6				
0xffc7				

Miss ratio =

(ii) Assuming that the cache is organized as **four-way set associative** with **2-byte block size**, determine the number of bits in the offset, index and tag fields. Starting with an empty cache, show the offset, index and tag (**in binary**) for each address reference in the list and indicate whether it is a hit or a miss. Assume that a FIFO replacement policy is used. What is the miss ratio for this sequence on this cache?

Offset = Index = Tag=

Address	Tag	Index	Offset	Hit/Miss
0x00c2				
0x00c3				
0x00c4				
0x00c5				
0x0ec2				
0x0ec3				
0x0ec4				
0x0ec5				
0x00c2				
0x00c3				
0x00c4				
0x00c5				
0x00c6				
0x00c7				
0xffc6				
0xffc7				

Miss ratio =

[12 Points]

(Q6) A processor runs at 2 GHz and has a CPI=1.4 for a perfect cache (i.e. without including the stall cycles due to cache misses). Assume that load and store instructions are15% of the instructions. The processor has an I-cache with a 4% miss rate and a D-cache with 6% miss rate. The hit time is 1 clock cycle. Assume that the time required to transfer a block of data from the RAM to the cache, i.e. miss penalty, is 40 ns.

(i) What is the number of stall cycles per instruction and the overall CPI?

(ii) What is the average memory access time (AMAT) in ns?

(iii) Discuss how you can reduce the AMAT.